

LotionChess

Brukermanual

Innhold	
Generell informasjon	1
Oppstart.....	1
Valgmeny for spilltype	1
«Singel windowed network game»	2
«Double-windowed localhost game»	3
Brukergrensesnitt	3
Statusbar:	3
Logg og chat:	4
Sjakkbrett:	5
Avslutt spillet	5

Generell informasjon

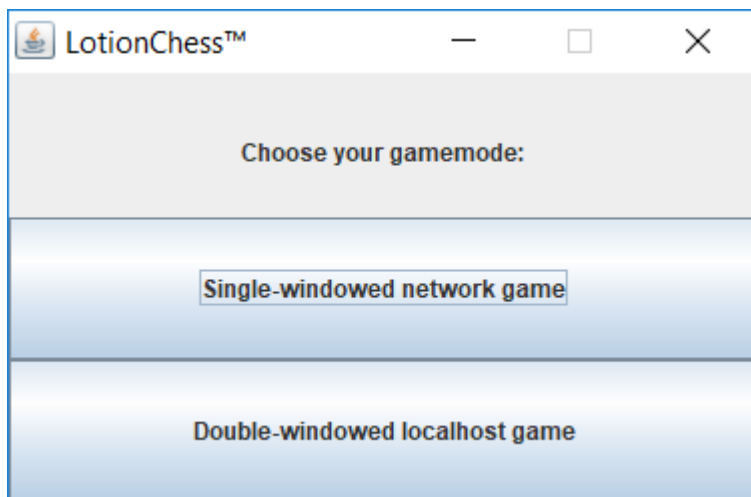
Denne brukermanualen er ment for å gi tilstrekkelig informasjon i bruk av LotionChess(LC) for nye brukere og gir en innføring i brukergrensesnitt og funksjonalitet. Brukermanualen forklarer ulike aspekter ved LC med bruk av illustrasjoner fra det faktiske spillet samt forklarende tekst. Det forutsettes at nye spillere er kjent med regler for sjakk. Reglene for sjakk finnes på denne linken: <https://no.wikipedia.org/wiki/Sjakkregler>

OBS! Merk at hvis du ønsker å kjøre Chess.jar så trenger den samtlige undermapper i mappen sin. Undermappene er: /src, /img og /stockfish-9-win. /src mappen trengs grunnet /src/config/config.xml.

Oppstart

Valgmeny for spilltype

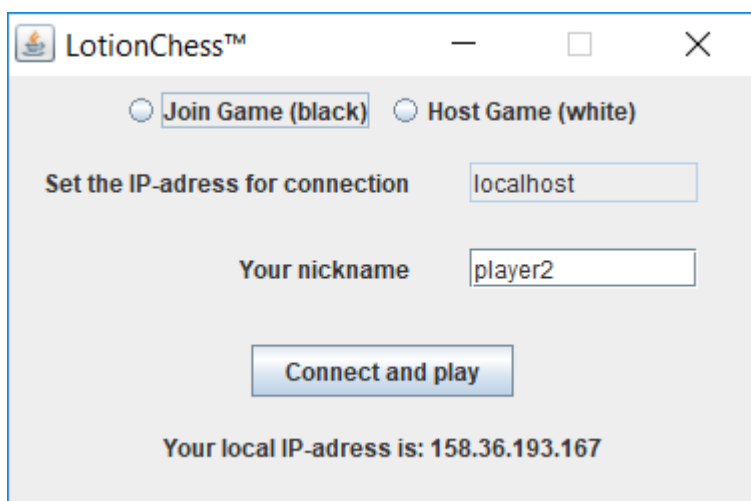
Når du åpner spillet for første gang vil du få opp et vindu som gir deg to alternativer. Du kan velge mellom «Singel windowed network game» eller «Double-windowed localhost game»



«Singel windowed network game» gir deg mulighet til å spille LC over nett med andre personer og «Double-windowed localhost game» gir deg mulighet til å spille LC med to personer på samme datamaskin.

«Singel windowed network game»

Hvis du har valgt alternativt for enkelt vindu, så vil du få opp en ny meny som gir deg mulighet til enten å “hoste” eller “joine” et spill. Hvis du “hoster” er det maskinen din som er vertskap for spillet, hvis du “joiner” kobler du deg til en annen maskin som “hoster”.



Hvis du velger “Join Game (black)” vil feltet for IP-adresse åpne seg og du vil nå få muligheten til å skrive inn en IP-adresse. Denne IP-adressen må samsvare med IP-adressen til maskinen som “hoster”. Skriv inn ditt ønskede nickname og klikk “Connect and play”.

Hvis du ønsker å “hoste” et spill så velger du “Host Game (white)”, skriver inn ønsket brukernavn og deretter klikk “Connect and play”.

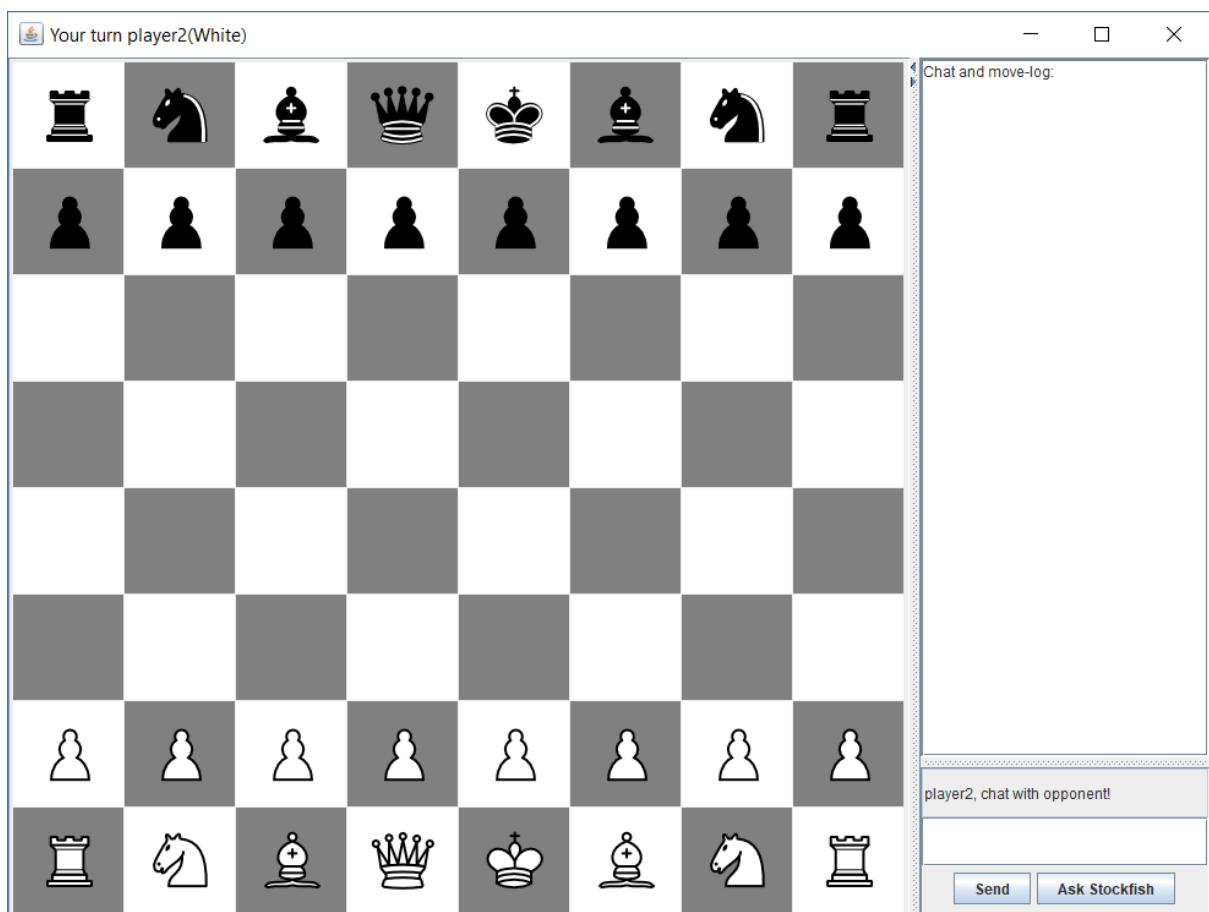
NB! For å kunne starte et spill over nett må “host” starte spillet først.

«Double-windowed localhost game»

Hvis du velger dette alternativet vil du få opp to vinduer på skjermen som viser sjakkbrettene til hver av spillerne. Spillet er dermed klart for å spilles med en gang. Denne modusen gjelder om du ønsker å spille med motspiller på samme datamaskin.

Brukergrensesnitt

Når du har gjennomført oppstart av LC så vil du få opp følgende vindu(er):



Statusbar:



Denne baren gir status på hvem sin tur det er, enten det er din eller motstanders tur. Baren gir også mulighet til å minimere og lukke spillet ved hjelp av knappene til høyre.

Logg og chat:

Chat and move-log:
Spiller 2(Black): Hello!
Spiller 1(White): Hi!
Spiller 2(Black): Ready?
Spiller 1(White): Yes!

Spiller 2, chat with opponent!

Send

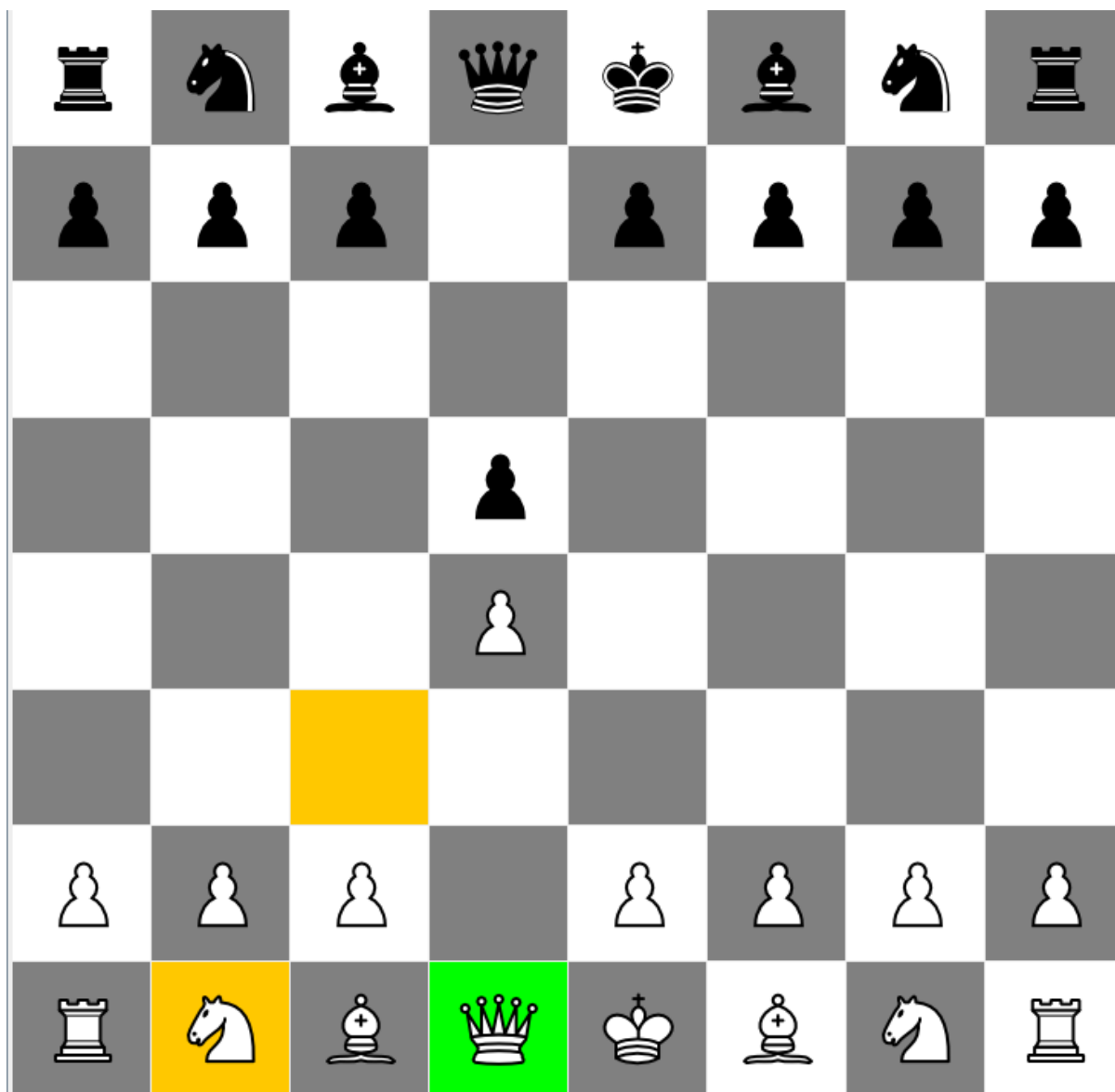
Ask Stockfish

Denne delen er sammensatt av en logg og en chat. I den hvite boksen øverst får du opp meldinger i form av beskjeder som blir sendt mellom spillere. Her vil også trekk som er blitt gjort stå oppført.

Hvis man ønsker å sende beskjeder kan man gjøre dette ved å skrive i den nederste boksen for så å trykke "Send" eller klikke på enter-tasten(↵) på tastaturet.

Hvis man ønsker å få et trekk som er anbefalt av Stockfish, som neste beste trekk, kan man gjøre dette ved å klikke på «Ask Stockfish»

Sjakkbrett:



Når du klikker på en brikke for å gjøre et trekk, vil firkanten brikken står i bli markert med en **grønn** farge og du kan dermed klikke på feltet du vil flytte brikken til. De **oransje** fargene er en indikasjon på hvilket trekk som er best. Denne anbefalingen er gjort av Stockfish-motoren (<https://stockfishchess.org/>) for sjakk. I dette tilfellet er det beste trekket å flytte hesten til feltet med **oransje** farge. Denne anbefalingen får du ved å klikke «Ask Stockfish» som forklart over.

Avslutt spillet

Når du ønsker å avslutte spillet gjør du dette ved å klikke på krysset i høyre hjørne i statusbaren:



LotionChess

Teknisk manual

Innledning

I dette dokumentet skal vi forklare ulike høydepunkter av programkoden vi mener er spesielt interessant for dette prosjektet. Høydepunktene vil bli forklart med bruk av illustrasjoner (skjermbilder) og forklarende tekst.

Hvordan fungerer programmet?

Main-metoden starter en GameLobby som inneholder GameSetting og en GUI ved navn MainMenu. MainMenu inneholder det første menypanelet ved navn GameModeSelector. Ved valg av Single-windowed network game endres MainMenu til å vise menypanelet ved navn ConnectionMenu. Dette gir mulighet for å konfigurere ipadresse og kallenavn før man lager en instans av Game. Velger man derimot Double-windowed localhost game instansieres to Game hvor spillernavn hentes fra config.xml via GameSettings.

Hver instans av Game har ansvar for å holde orden på selve spill-logikken. Den oppretter et grafisk ChessGui som er en instans av Stockfish og en instans av NetworkServer eller NetworkClient for å koble seg til motstander sin instans av Game. Game holder også rede på historikk av trekk i List<move> moves som finner ut om det er spilleren sin tur og kan sjekke om et trekk er lovlig. Game tar imot motstandermeldinger fra nettverket i form av Action objekter som inneholder blant annet Log objekter (tekst som skal vises) og Move objekter (trekk som skal flyttes) og behandler disse deretter.

Høydepunkter

Stockfish-kommunikasjon (\src\Stockfish)

1. Innkapsling - eksempelvis isMoveLegal og sendCommand

```
public boolean isMoveLegal(String movesString, Move move)
{
    boolean answer = false;
    String moveString = move.toString();
    String fen = "";
    String newFen = "";

    try {
        this.stockFishWriter.write(StockfishCommands.START_NEW_GAME);
        this.stockFishWriter.flush();
        this.stockFishWriter.write(StockfishCommands.SET_POSITION +
```

```

movesString + "\n");
    this.stockFishWriter.flush();
    fen = this.sendCommand(StockfishCommands.GET_POSITION,
StockfishReturns.FEN_POSITION);
    this.stockFishWriter.write(StockfishCommands.START_NEW_GAME);
    this.stockFishWriter.flush();
    this.stockFishWriter.write(StockfishCommands.SET_POSITION +
movesString + " " + moveString + "\n");
    this.stockFishWriter.flush();
    newFen = this.sendCommand(StockfishCommands.GET_POSITION,
StockfishReturns.FEN_POSITION);
}
catch (IOException e) {
    e.printStackTrace();
}
fen = translateAnswerToFen(fen);
newFen = translateAnswerToFen(newFen);
if(!newFen.equals(fen)) {
    answer = true;
}
return answer;
}

```

2. Generering av fen-string. Setter posisjon og henter posisjon.

```

public String getFen()
{
    if (this.rowHasZeroPieces()) {
        return "8/";
    }

    StringBuilder fen = new StringBuilder();
    int emptyField = 0;
    for(Field field : fields) {
        if(field.hasPiece()) {
            fen.append(emptyField == 0 ? field.getCurrentPieceName() :
emptyField + field.getCurrentPieceName());
            emptyField = 0;
        }
        else {
            emptyField++;
        }
    }
    if (emptyField != 0){
        fen.append(emptyField);
    }
}

```

```
    return fen + "/";  
}
```

ChessPiece-klassene (\src\GamePieces)

1. Innkapsling - bruk av abstrakt klasse for delt funksjonalitet mellom alle brikkene

```
package GamePieces;  
  
public class Bishop extends ChessPiece {  
  
    public Bishop(boolean white) {  
        super("b", white);  
        if(this.isWhite()) {  
            setImagePath("Chess_blt60.png");  
        }  
        else {  
            setImagePath("Chess_bdt60.png");  
        }  
    }  
}
```

Spill over IP (\src\Helpers\Action.java)

1. All kommunikasjon via nettverket bruker samme objekt med individuell last (Action).
Eksempelvis fra fil: src\Helpers\NetworkServer.java og
src\Helpers\NetworkClient.java

```
public void onNewAction(Action action)  
{  
    try {  
        this.oos.writeObject(action);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Bruk av interfaces (\src\Interfaces)

1. Løs kobling mellom hovedelementene ved bruk av interfaces
2. Egne lytter-klasser for større fleksibilitet.

GUI (\src\GUI)

1. Inndeling av forskjellige områder (JSplitPane) - Eget område for sjakkbrettet, chat og logg.
2. Brukervennlighet - eksempelvis knapper som blir aktivert og deaktivert ved behov og inputs som reagerer på enter-tasten og egen "send"-knapp.
3. Fremheving - markerer valgt brikke (grønn) og beste trekk (oransje) ved å trykke på en knapp som spør stockfish om beste trekk.

Konfigurasjonsfil

1. Egen konfigurasjonsfil som laster inn sist brukte innstillinger ved valg av nickname og IP i startmenyen. Filen er skrevet i XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment/>
<entry key="localPlayerBlack">Spiller 2</entry>
<entry key="localPlayerWhite">Spiller 1</entry>
<entry key="ip">localhost</entry>
<entry key="nickName">player2</entry>
</properties>
```

Bruk av konfigurasjonsfilen skjer i: \src\Config\GameSettings.java

Logg

1. Allokert ansvar for logging av handlinger. Inkludert oppretting av logg-objekter. Følgende logg-objekter opprettes: Chat, trekk, ugyldige trekk og bruk av Stockfish-hjelp.

Beste trekk (\src\Helpers\Game.java)

1. Henter fen-streng basert på spillets tilstand og setter posisjonen i Stockfish. Den sender så kommandoen go() og oversetter svaret til et move-objekt. Dermed forteller metoden GUI at den skal fremheve dette trekket på brettet (oransje markør).

```
private Move getBestMoveFromStockfish()
{
    String fen = buildCurrentFen();
    System.out.println("FEN position: " + fen);
    stockFish.setFEN(fen);
    System.out.println("Spør stockfish om bestmove");
    String moveString = this.stockFish.getComputerMoveByFen();
    System.out.println("Stockfish.Stockfish svarte: " + moveString);
}
```

```
return this.translateFromMoveString(moveString);  
}
```

Denne koden viser også god bruk av semantiske navn. Eksempelvis:
“this.getComputerMoveByFen();”

Utfordringer

Logikk for reglene i sjakk

Per dags dato er det noen av sjakkreglene som ikke er implementert i produktet. Disse er blant annet:

- Bondekonvertering
- En passant
- Rokade
- Sjakkmatt
- Sjakk
- Remis

Utenom dette fungerer sjekk for gyldige/ugyldige trekk.

Andre utfordringer

Utfordringer i forhold til nettverk har også vært en del av prosessen i løsningen av oppgaven. Disse innebærer blant annet logikk for å sjekke om en spiller har koblet til vertsmaskinen slik at spilleren på vertsmaskinen ikke får gjort et trekk før det har blitt koblet til en spiller.